Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2023.033585
Article

Tech Science Press

Automated Disabled People Fall Detection Using Cuckoo Search with Mobile Networks

Mesfer Al Duhayyim*

Department of Computer Science, College of Sciences and Humanities-Aflaj, Prince Sattam bin Abdulaziz University, Saudi Arabia
*Corresponding Author: Mesfer Al Duhayyim. Email: m.alduhayyim@psau.edu.sa

Received: 21 June 2022; Accepted: 26 October 2022

Abstract: Falls are the most common concern among older adults or disabled people who use scooters and wheelchairs. The early detection of disabled persons' falls is required to increase the living rate of an individual or provide support to them whenever required. In recent times, the arrival of the Internet of Things (IoT), smartphones, Artificial Intelligence (AI), wearables and so on make it easy to design fall detection mechanisms for smart homecare. The current study develops an Automated Disabled People Fall Detection using Cuckoo Search Optimization with Mobile Networks (ADPFD-CSOMN) model. The proposed model's major aim is to detect and distinguish fall events from non-fall events automatically. To attain this, the presented ADPFD-CSOMN technique incorporates the design of the MobileNet model for the feature extraction process. Next, the CSO-based hyperparameter tuning process is executed for the MobileNet model, which shows the paper's novelty. Finally, the Radial Basis Function (RBF) classification model recognises and classifies the instances as either fall or non-fall. In order to validate the betterment of the proposed ADPFD-CSOMN model, a comprehensive experimental analysis was conducted. The results confirmed the enhanced fall classification outcomes of the ADPFD-CSOMN model over other approaches with an accuracy of 99.17%.

Keywords: Disabled people; human-computer interaction; fall event detection; deep learning; computer vision

1 Introduction

Disabled people are highly susceptible to hazardous falls and are incapable of calling or getting help at crucial moments. During such emergency times, it is challenging for them to push the emergency button present in their necklace or the medical bracelet. This drawback creates a need for the development of a medical alert mechanism that remains helpful in the detection of falls [1]. A gadget should be developed so that it utilizes the technology to predict the fall events of its wearer. In case of such a prediction, the device should automatically send an alert to the healthcare monitoring company. Fall can be defined as any unanticipated event that leads the individual to rest in a low phase [2]. Falls cause injuries that might be fatal most of the time, while they also result in psychological complaints too. Those individuals who met fatal falls may not be able to perform their day-to-day activities and suffer from constraints, anxiety, fear and depression of falling again. The main biological problem among elder adults is the fear of falling and limiting their Activities of Daily Life (ADL) [3]. This fear further leads to activity constraints which in turn weakens the muscles and causes insufficient gait balance. Consequently, the independence of the individual gets affected, and old-age people rely on others for their mobility. In this background, wearable or remote technologies are required for tracking, detecting and preventing falls from enhancing their overall Quality of Life (QoL). Falls' understanding can be divided into fall prevention and detection [4].

When a fall is detected with the help of cameras or sensors to summon help, it is called fall detection [5]. Conversely, fall prevention focuses on averting falls by monitoring the locomotion of the human beings. Various advanced mechanisms have been developed in recent years with the help of sensors and methods to detect as well as prevent falls [6]. Fall detection refers to the task of identifying a specific object or a specific incident in the provided context. On the other hand, fall recognition can be explained as the arrangement of membership of the incident or an instance for a specific class. Thus, the fall 'detection' process can be defined as an identification process that determines whether a fall has occurred from the presented data. Fall 'recognition' can be defined as the recognition of a particular kind of fall event, such as a fall from the sitting position, forward fall, fall from a lying position, backward fall, fall from the standing position, and so on [7,8]. Fall recognition is highly helpful in increasing the appropriateness of the response. For instance, a fall from a standing position is fatal, whereas the outcomes of a fall from either lying or sitting posture depend on the prevailing conditions. If the fall type is appropriately familiar, effectual responses can be raised to handle the probable complexities of the particular kind of fall [9]. This study does not openly refer to whether the reviewed mechanism is either a fall recognition or a fall detection mechanism. Rather, it uses the word 'fall detection' to represent the general fall detection and activity recognition mechanisms. The Machine Learning (ML) technique offers a learning capability to the mechanism on the basis of trends and datasets in the data. At the time of the data collection process, the sensors collect the data related to distinct fall variables [10]. Therefore, ML techniques are utilized in the data processing mechanisms for identification and the classification of fall activities based on their application needs.

The current study presents an Automated Disabled People Fall Detection using Cuckoo Search Optimization with Mobile Networks (ADPFD-CSOMN) model. The presented ADPFD-CSOMN technique incorporates the MobileNet model for the feature extraction process in the preliminary stage. In addition, the CSO algorithm is exploited for optimal modification of the hyperparameters involved herewith. Finally, the Radial Basis Function (RBF) classification model recognises and classifies an instance as either a fall event or a non-fall event. In order to validate the supremacy of the proposed ADPFD-CSOMN model, a comprehensive experimental analysis was conducted.

2 Literature Review

Yacchirema et al. [11] developed the Internet of Things and Ensemble (IoTE)-Fall method, a smart fall detection technology that can be used among elderly persons in the indoor environment. The proposed technique used the IoT technology and the ensemble ML method. The IoTE-Fall technique applied a 3D-axis accelerometer embedded with IPv6 over Low-Power Wireless Personal Area Network (6lowpan) wearable device. This device can capture real-time data regarding the movements of elderly volunteers. In the study conducted earlier [12], a low-cost extremely-precise Machine Learning (ML)-related fall detection technique was presented. Especially a novel online feature extraction methodology was developed in this study that proficiently applied the time features of the falls. Additionally, a new ML-based technique was developed to achieve the numerical complexity trade-off or optimal accuracy. The low computation cost incurred by the presented approach allowed it to be embedded in wearable sensors.

Further, the power requirement was comparatively less than the rest of the devices. These features improve the independence of wearable devices. On the other hand, the need for battery replacement or recharge gets reduced. Hassan et al. [13] presented a fall detection architecture that identifies the falls of elderly persons and assists the family members as well as their caregivers by instantly localizing the victims. In this work, the information retrieved from the accelerometer sensors on smartphones was analysed and processed through online fall detection techniques installed in smartphones. This technique transmits an indoor sound alert to the family member via a wireless access point at home or through Short Message Service (SMS) in case of outdoor to the caregiver or the hospital via mobile network BS. A hybrid DL method was applied in this study for the fall detection process.

Zerrouki et al. [14] developed an advanced technique for consistent identification of the falls related to the human silhouette shape variance in vision monitoring. This mission was accomplished based on the steps given herewith (i) Reduction of the feature vector dimension via a differential evolution method; (ii) Presentation of a curve-let transform along with the area ratios to recognize the postures of a human being in the image; (iii) Adaptation of Hidden Markov Mechanism for the classification of a video sequence as either non-fall or fall actions (iv) and identification of the postures using the Support Vector Machine (SVM) model. Maitre et al. [15] recommended a Deep Neural Network (DNN) method using three ultra-wide band radars to distinguish the fall events in an apartment measuring 40 square meters. The Deep Neural Network had a Convolutional Neural Network (CNN) that was stacked with Long Short Term Memory (LSTM) model and a Fully Convolutional Network (FCN) to recognize the fall events. In other terms, the problem tackled here was a binary classifier attempt to discriminate the fall events from the non-fall events.

3 Process Involved in Fall Detection Model

In this study, a novel ADPFD-CSOMN method has been developed to identify and classify fall events. The presented ADPFD-CSOMN technique incorporates the MobileNet model for the feature extraction process during the preliminary stage. In addition, the CSO technique is exploited for optimal fine-tuning of the hyperparameters involved. At last, the RBF classification algorithm is utilized to recognize and classify an event as either fall or a non-fall event. Fig. 1 illustrates the overall block diagram of the ADPFD-CSOMN approach.

3.1 Feature Extraction Using MobileNet Model

At first, the presented ADPFD-CSOMN technique incorporates the MobileNet model design for feature extraction. The MobileNet structure is effective with less number of features in the name of Palmprint detection [16]. Further, the MobileNet framework is depth-wise. The basic architecture depends on different abstraction layers, whereas the mechanism behind the distinct convolution layers seems to be a quantized formation that evaluates the common challenge. The difficulty of 1 × 1 layer is named point-wise difficulty, whereas the platform to make a detailed abstraction layer lies within this in-depth structure. It points over a typical Rectified Linear Unit (ReLU). Then, a parameter ω is added to the resolution multiplier to minimize the dimensions of the input images. Each layer internally demonstrates to have a comparable parameter. The feature vector map has a size of $F_m \times F_m$, and the filter is of size, $F_s \times F_s$. Further, the input variable is denoted by p, whereas the output variable is identified by q. For the core abstract layer, the computation effort is considered ce, and the parameter is evaluated using the subsequent formula.

$$c_e = F_s \cdot F_s \cdot \omega \cdot \alpha F_m \cdot \alpha F_m + \omega \cdot \rho \cdot \alpha F_m \cdot \alpha F_m \tag{1}$$

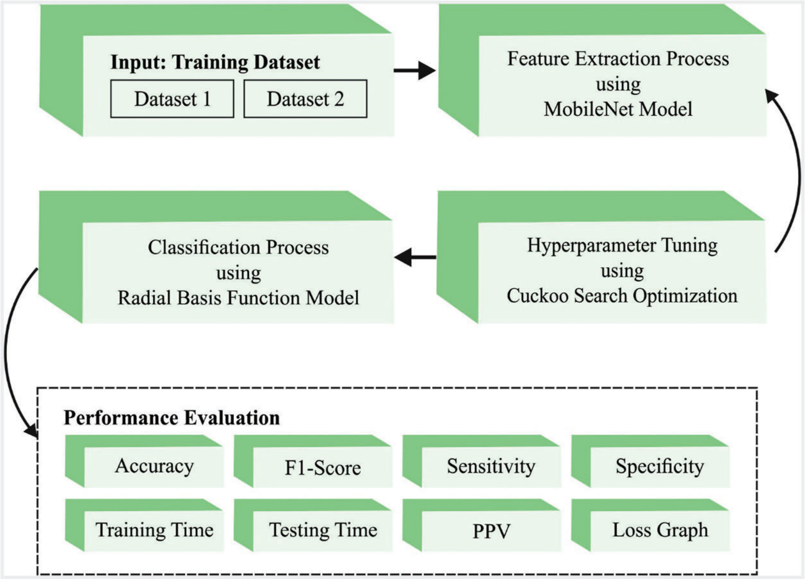
Input: Training Dataset
Dataset 1
Dataset 2
Feature Extraction Process using MobileNet Model
Classification Process using Radial Basis Function Model
Hyperparameter Tuning using Cuckoo Search Optimization
Performance Evaluation
Accuracy
F1-Score
Sensitivity
Specificity
Training Time
Testing Time
PPV
Loss Graph

Figure 1: Overall block diagram of the ADPFD-CSOMN approach

Since the multiplier value is a context-specific entity, it is also used in the classification of skin diseases. Here, the value of the multiplier ω is regarded as 1 to n. The value of the parameter resolution multiplier known as α is assumed to be 1. The computation effort is known as $cost_e$, and this parameter is evaluated using the formula given below.

$$cost_e = F_s \cdot F_s \cdot \omega \cdot \rho \cdot F_m \cdot F_m \tag{2}$$

The projected algorithm incorporates both depth-wise and point-wise convolution layers, which are limited by a renowned depletion parameter using the d parameter and are evaluated as given below.

$$d = \frac{F_s \cdot F_s \cdot \omega \cdot \alpha F_m \cdot \alpha F_m + \omega. \cdot \rho \cdot \alpha F_m \cdot \alpha F_m}{F_s \cdot F_s \cdot \omega \cdot \rho \cdot F_m F_m} \tag{3}$$

Two hyper features, such as the resolution and the width multiplier, help in adjusting the feasible size window to achieve a precise prediction. In this work, the input size of the image is 224 × 224 × 3. The two initial values, i.e., (224 × 224), correspond to the height and width of the image. The value must be above 32. The third value indicates that the image has three input channels. The projected structure contains 32 filters, whereas the filter size is 3 × 3 × 3 × 32. The MobileNet model substitutes a complex convolution layer. The layer has a convolution layer sized 3 × 3. The input dataset is buffered along with a pointwise convolutional layer sized 1 × 1. This is inclusive of the filtered parameter to construct a novel component. The concept above is to shorten the mechanism and make it faster than the standard convolution method.

3.2 Hyperparameter Adjustment Using CSO Algorithm

In this stage, the CSO algorithm is exploited for optimal fine-tuning of the hyperparameters involved herewith. The CSO algorithm is inspired by the social performance of Cuckoo [17]. This algorithm is a population-related search method in which an optimal outcome is identified for the enhanced problem. Computationally, the CSO approach is formulated as two fundamental stages. In the initial stage, the accessibility of the host nest is fixed. The following posit is that every Cuckoo finds only an individual egg at timestamp t to arbitrarily choose the host nest. At last, the maximum quality egg from the optimal nest proceeds to the upcoming generation. Consider that $X_i(t)$ indicates the existing searching region of the Cuckoo, where $i = 1, 2, \cdots, N$ at t time, characterized by $X_i = \left(x_i^1, x_i^2, \cdots\cdots, x_i^n\right)$ from n dimensional quandary. Next, the initialized searching region $X_i(t+1)$ to the future generation at $t+1$ time is mathematically evaluated using the following equation.

$$X_i(t+1) = X_i(t) + \alpha Levy(\lambda) \tag{4}$$

If $\alpha > 0$ indicates the step-size, which is comparable to the scale of the quandary of the curiosity, the highest case α is usually assumed as 1. It provides a random step in which the walk is derived in Levy distribution for a vast number of colossal steps that are illustrated using the following equation:

$$Levy\ (\lambda)u = t^{-\lambda}, \quad where\ \ 1 < \lambda < 3 \tag{5}$$

Mantegna's and McCulloch's techniques are two popular techniques used in this optimization method. The Levy step size is calculated by following the Mantegna approach as given herewith.

$$Levy = \frac{v}{|v|^{1/(\lambda-1)}} \tag{6}$$

This condition is a forsaken probability, the whole size of the population and the maximum quantity of cuckoo reproduction during its lifetime is set as the utilizer; however, the first term at the initial stage is accomplished. Fig. 2 demonstrates the flowchart of the CSO technique.

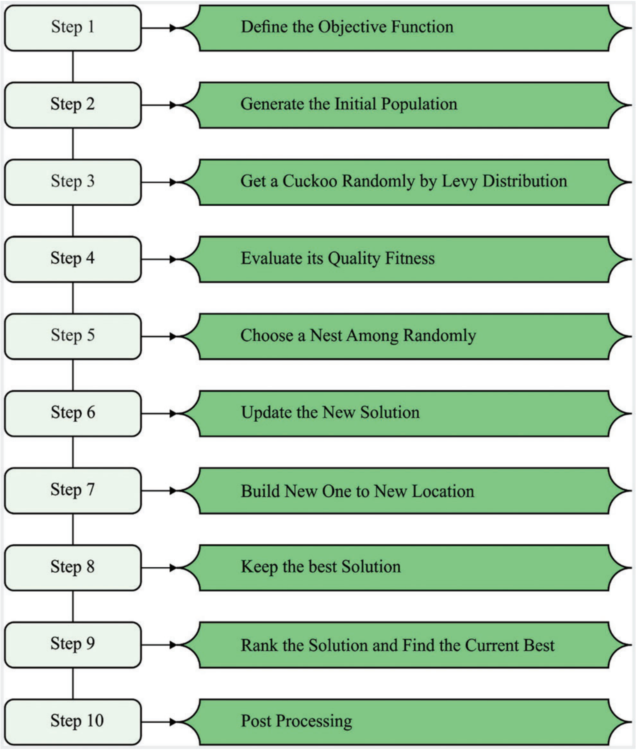
Step 1
Define the Objective Function
Step 2
Generate the Initial Population
Step 3
Get a Cuckoo Randomly by Levy Distribution
Step 4
Evaluate its Quality Fitness
Step 5
Choose a Nest Among Randomly
Step 6
Update the New Solution
Step 7
Build New One to New Location
Step 8
Keep the best Solution
Step 9
Rank the Solution and Find the Current Best
Step 10
Post Processing

Figure 2: Flowchart of CSO technique

Consider that t indicates the existing generation, t_{max} denotes the maximum span of the animals with life redundancy and the impact of the cuckoo that roams from the lifetime. As a result, the n-dimensional problem of the cuckoo for initial generation $t = 1$ is set-up as characterized in the following expression.

$$x_1^n(t=1) = randn \times (Upper^n - Lower^n) \tag{7}$$

In Eq. (7), $Lower^n$ and $Upper^n$ denote the lower and the upper limits of the searching region of *n-th* feature. This ability to maintain the time in the optimization method stays in a single perimeter.

In Eq. (6), υ and v demonstrate the time taken during standard distribution. In other terms, $\upsilon \sim N\left(0,\ \delta^2\right)$ and $v \sim N(0,\ 1)$ are formulated as given below.

$$\delta = \left(\frac{\Gamma(1+\beta)\sin\left(\frac{\pi\pi}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right)\beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}} \tag{8}$$

In Eq. (8), Γ denotes the gamma function as denoted herewith.

$$\Gamma(p) = \int_0 e^{-t} t^{p-1} dt \ \& \ \beta \in [0,\ 2] \ is\ a\ scale\ factor. \tag{9}$$

CSO method derives a fitness function to achieve enhanced classification performance. It sets a positive digit to denote the superior outcomes of the candidate solutions. In this study, the reduced classification error rate is taken as the fitness function, as shown in Eq. (10). The finest solution contains less error rates, whereas the poorest solution has a high error rate.

$$fitness(x_i) = ClassifierError\,Rate(x_i) = \frac{number\ \ of\ \ misclassified\ \ samples}{Total\ \ number\ \ of\ \ samples} * 100 \tag{10}$$

3.3 Fall Detection Using RBF Model

In this final stage, the Radial Basis Function (RBF) classification method is utilized to recognize and classify an instance as either a fall event or a non-fall event. RBF is a well-known supervised Neural Network (NN) learning model [18] that falls under the MLP network type. The RBF network is established through three subsequent layers given below.

Input Layer. It transfers the input with no distortion

RBF Layer. Hidden layer that comprises RBF

Output Layer. Simple layer that comprises a linear function.

Normally, the basis function takes the form $\phi = ||\vec{x}_i - \vec{\mu}_i||$, in which the function depends on the distance between the input vectors, $\vec{x}$ and $\vec{\mu}$. Gaussian function is the commonly-applied basis function.

$$\phi = \ \exp \frac{||\vec{x}_i - \vec{\mu}_i||}{2\sigma_j^2}, \tag{11}$$

In Eq. (11), $\vec{\mu}_i$ describes the centre of the basis function, whereas σ_j denotes the width variable that controls the spread of the curve. Usually, the centre is designated either by a fuzzy or a non-fuzzy clustering algorithm. Here, the K-means algorithm is applied to select the first cluster centre during the initial phase. At later stages, the centre is additionally fine-tuned through point symmetry distance measure. The number of neurons in the output layer is equivalent to the class count of the classification issue. All the output neurons compute a linear weighted amount of the output of the hidden neuron in the following equation.

$$y_i(x) = \sum\nolimits_{i=1}^{N} \varphi_i(X) \cdot w_i. \tag{12}$$

The weight vector is described by minimizing the mean squared variances among the classification output.

$$y_k = \sum\nolimits_{j=0}^{m} w_{k,\mathrm{j}} s_i. \tag{13}$$

Then, the target value t_k is given by the following equation.

$$E = \frac{1}{2}\sum\nolimits_{k=1}^{M} (y_k - t_k)^2 \tag{14}$$

The parameters $(\Delta w, \ \ \Delta\mu, \ \ \Delta\sigma)$ are represented as follows.

$$\frac{\partial E}{\partial w_{ki}} = \frac{\partial E}{\partial y_k}\frac{\partial y_k}{\partial w_{ki}} \tag{15}$$

or

$$\frac{\partial E}{\partial y_k} = (t_k - y_k). \tag{16}$$

Hence,

$$\frac{\partial E}{\partial w_{ki}} = -(t_k - y_k)s_i. \tag{17}$$

After calculations, the following notations are attained.

$$\frac{\partial E}{\partial \mu_{ji}} = \sum_k \frac{\partial E}{\partial y_k}\frac{\partial y_k}{\partial s_j}\frac{\partial s_j}{\partial \mu_{ji}} = \frac{s_j}{\sigma_j^2}\left(x_i - \mu_{ji}\right)\sum\nolimits_{i=1}(t_k - y_k)w_{kj}, \tag{18}$$

$$\frac{\partial E}{\partial \sigma_{ji}} = \sum_k \frac{\partial E}{\partial y_k}\frac{\partial y_k}{\partial s_j}\frac{\partial s_j}{\partial \sigma_{ji}} = \frac{2s_j}{\sigma_j^2}\ log\ s_j \sum_{i=1}(t_k - y_k)w_{kj},$$

4 Experimental Validation

In this section, the proposed ADPFD-CSOMN model was experimentally validated using two datasets, namely, the Multiple Cameras Fall (MCF) dataset (available at http://www.iro.umontreal.ca/~labimage/Dataset/) and UR Fall Detection (URFD) dataset (available at http://fenix.univ.rzeszow.pl/~mkepski/ds/uf.html).

Table 1 and Fig. 3 show the fall detection results attained by the proposed ADPFD-CSOMN algorithm on MCF dataset. The results infer that the ADPFD-CSOMN method achieved an effectual recognition performance under each class. For example, on run-1, the ADPFD-CSOMN model achieved $accu_y$, Positive Predictive Value (PPV), $sens_y$, $spec_y$ and $F1_{score}$ values such as 99.59%, 99.03%, 99.12%, 99.40% and 99.52% respectively. In addition, on run-3, the proposed ADPFD-CSOMN approach accomplished $accu_y$, PPV, $sens_y$, $spec_y$ and $F1_{score}$ values such as 99.59%, 99.48%, 99.98%, 99.18% and 99.39% correspondingly. Along with that, on run-5, the presented ADPFD-CSOMN technique obtained $accu_y$, PPV, $sens_y$, $spec_y$ and $F1_{score}$ values such as 99.63%, 99.41%, 99.02%, 99.36% and 99.48% correspondingly.

Table 1: Analytical results of the ADPFD-CSOMN approach under different measures on MCF dataset

Multiple cameras fall dataset					
No. of runs	Accuracy	PPV	Sensitivity	Specificity	F1-Score
Run-1	99.59	99.03	99.12	99.40	99.52
Run-2	99.64	99.50	98.91	99.07	99.21
Run-3	99.59	99.48	98.98	99.18	99.39
Run-4	99.05	98.95	99.43	99.03	99.00
Run-5	99.63	99.41	99.02	99.36	99.48
Average	99.50	99.27	99.09	99.21	99.32

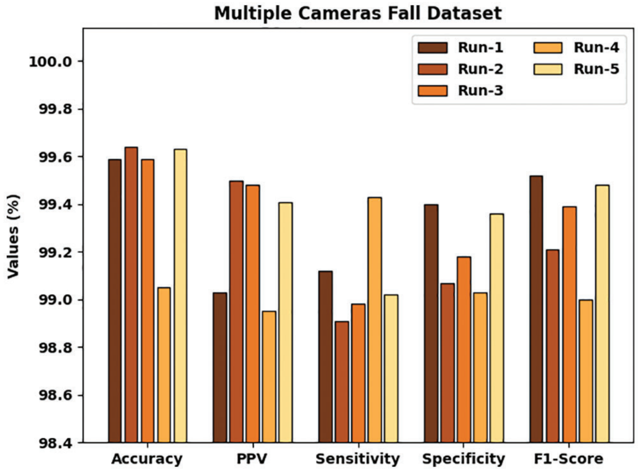

Figure 3: Analytical results of the ADPFD-CSOMN approach on MCF dataset

Fig. 4 demonstrates the average fall detection outcomes accomplished by the proposed ADPFD-CSOMN algorithm on test MCF dataset. The figure implies that the proposed ADPFD-CSOMN method reached average $accu_y$, PPV, $sens_y$, $spec_y$ and $F1_{score}$ values such as 99.50%, 99.27%, 99.09%, 99.21% and 99.32% respectively.

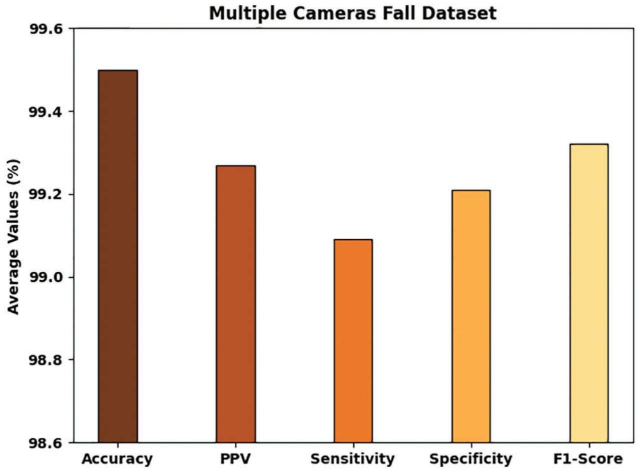

Figure 4: Average analytical results of the ADPFD-CSOMN approach on MCF dataset

Both Training Accuracy (TA) and Validation Accuracy (VA) values, obtained by the ADPFD-CSOMN method on MCF dataset, are shown in Fig. 5. The experimental outcomes revealed that the presented ADPFD-CSOMN system reached superior TA and VA values whereas the VA values were higher than the TA values.

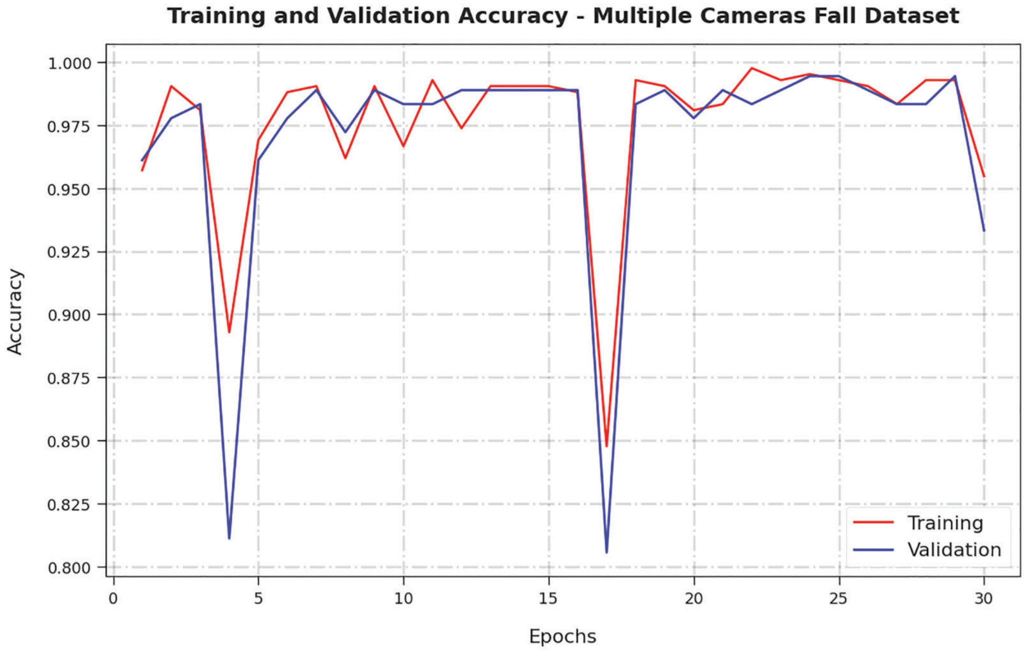

Figure 5: TA and VA analyses results of the ADPFD-CSOMN approach on MCF dataset

Both Training Loss (TL) and Validation Loss (VL) values, acquired by the ADPFD-CSOMN approach on MCF dataset, are exhibited in Fig. 6. The experimental outcomes denote that the proposed ADPFD-CSOMN approach exhibited the least TL and VL values while the VL values were lesser than the TL values.

Table 2 and Fig. 7 portray the brief overview of the comparative analysis results, attained by the proposed ADPFD-CSOMN method and other recent approaches on MCF dataset [19]. The figure reports that the 1D-CNN, 2D-CNN, ResNet-101 and the ResNet-50 models reported the least $accu_y$ values such as 94.27%, 95.44%, 96.31% and 95.96% respectively. In the meantime, the depthwise model, VGG-16 and the VGG-19 methods produced reasonable $accu_y$ values such as 97.85%, 97.87% and 98.20% respectively. However, the proposed ADPFD-CSOMN model gained the highest $accu_y$ of 99.50%.

The researchers conducted the Training Time (TRT) and Testing Time (TST) analyses in detail between the proposed ADPFD-CSOMN model and other recent models on MCF dataset and the results are shown in Table 3 and Fig. 8. Based on the TRT analysis outcomes, it can be inferred that the proposed ADPFD-CSOMN model achieved proficient results with a minimal TRT of 1043 s. But the depthwise technique, ResNet-101, ResNet-50, VGG-16, VGG-19, 1D-CNN and the 2D-CNN models produced high TRT values such as 1297, 3053, 2219, 2543, 2678, 3095 and 3228 s respectively. Also, based on the TST analysis outcomes, the ADPFD-CSOMN algorithm proved its proficiency with a minimal TRT of 657 s whereas the depthwise, ResNet-101, ResNet-50, VGG-16, VGG-19, 1D-CNN and the 2D-CNN models produced increased TST values such as 883, 1077, 800, 1485, 1784, 1423 and 1501 s correspondingly.

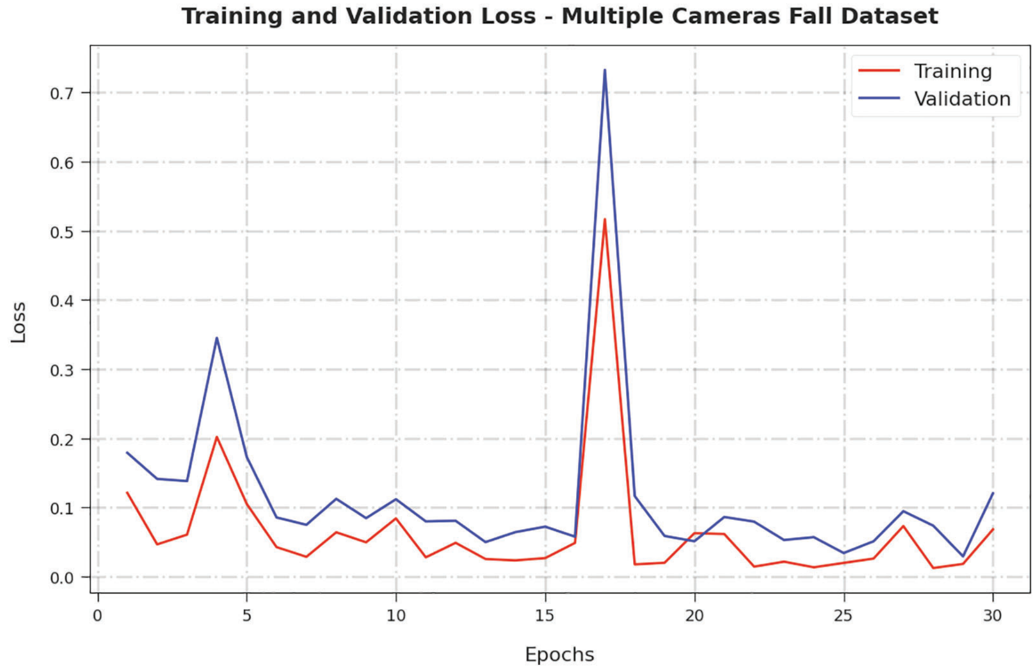

Figure 6: TL and VL analyses results of the ADPFD-CSOMN approach on MCF dataset

Table 2: Comparative analysis results of the ADPFD-CSOMN approach and other recent algorithms on MCF dataset

Methods	Accuracy (%)
ADPFD-CSOMN	99.50
Depthwise algorithm	97.85
ResNet-101 technique	96.31
ResNet-50 technique	95.96
VGG-16 algorithm	97.87
VGG-19 algorithm	98.20
1D-CNN model	94.27
2D-CNN model	95.44

Table 4 and Fig. 9 provide the fall detection results offered by the ADPFD-CSOMN model on URFD dataset. The results infer that the proposed ADPFD-CSOMN method displayed an effectual recognition performance for every class. For example, on run-1, the ADPFD-CSOMN model rendered $accu_y$, PPV, $sens_y$, $spec_y$ and $F1_{score}$ values such as 99.18%, 99.25%, 98.92%, 99.39% and 98.99% correspondingly. Moreover, on run-3, the proposed ADPFD-CSOMN model attained $accu_y$, PPV, $sens_y$, $spec_y$ and $F1_{score}$ values such as 99.05%, 99.01%, 99.14%, 99.08% and 99.12% correspondingly. Also, on run-5, the proposed ADPFD-CSOMN model yielded $accu_y$, PPV, $sens_y$, $spec_y$ and $F1_{score}$ values such as 99.20%, 99.02%, 98.98%, 99.18% and 99.42% correspondingly.

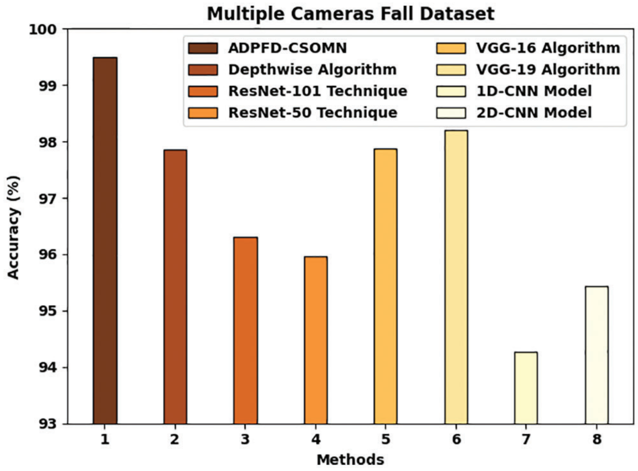

Figure 7: Comparative analysis results of the ADPFD-CSOMN approach on MCF dataset

Table 3: TRT and TST analyses results of the ADPFD-CSOMN approach and other recent algorithms on MCF dataset

	Time (sec)	
Methods	Training	Testing
ADPFD-CSOMN	1043	657
Depthwise algorithm	1297	883
ResNet-101 technique	3053	1077
ResNet-50 technique	2219	800
VGG-16 algorithm	2543	1485
VGG-19 algorithm	2678	1784
1D-CNN model	3095	1423
2D-CNN model	3228	1501

Fig. 10 establishes the average fall detection outcomes attained by the proposed ADPFD-CSOMN method on test URFD dataset. The figure implies that the ADPFD-CSOMN approach reached average $accu_y$, PPV, $sens_y$, $spec_y$ and $F1_{score}$ values such as 99.17%, 99.12%, 99.19%, 99.24% and 99.25% correspondingly.

Both TA and VA values, acquired by the ADPFD-CSOMN method on URFD dataset, are shown in Fig. 11. The experimental outcomes infer that the proposed ADPFD-CSOMN technique achieved the maximal TA and VA values while the VA values were higher than the TA values.

Both TL and VL values, reached by the proposed ADPFD-CSOMN approach on URFD dataset, are exhibited in Fig. 12. The experimental outcomes imply that the proposed ADPFD-CSOMN algorithm exhibited the minimal TL and VL values while the VL values were lesser than the TL values.

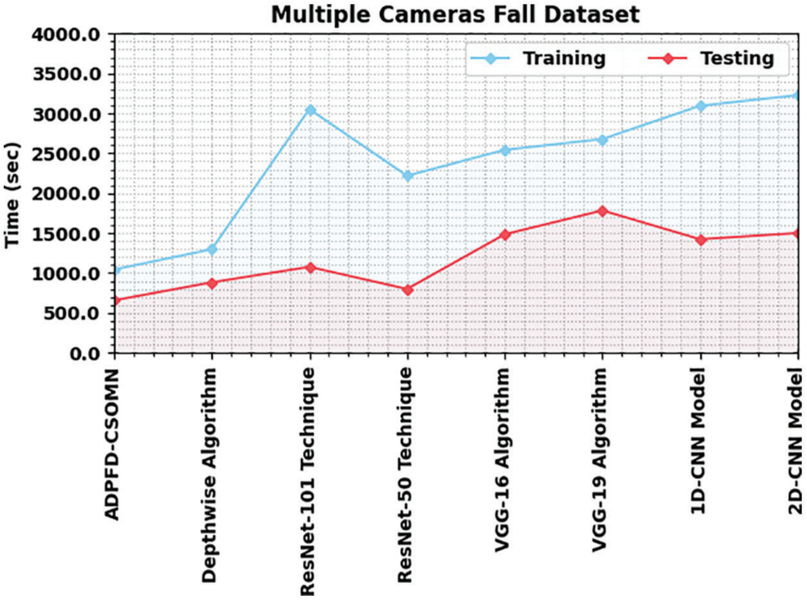

Figure 8: TRT and TST analyses results of the ADPFD-CSOMN approach on MCF dataset

Table 4: Analytical results of the ADPFD-CSOMN approach under different measures on URFD dataset

UR fall detection dataset					
No. of runs	Accuracy	PPV	Sensitivity	Specificity	F1-Score
Run-1	99.18	99.25	98.92	99.39	98.99
Run-2	99.01	99.13	99.51	99.06	99.22
Run-3	99.05	99.01	99.14	99.08	99.12
Run-4	99.41	99.20	99.38	99.49	99.50
Run-5	99.20	99.02	98.98	99.18	99.42
Average	99.17	99.12	99.19	99.24	99.25

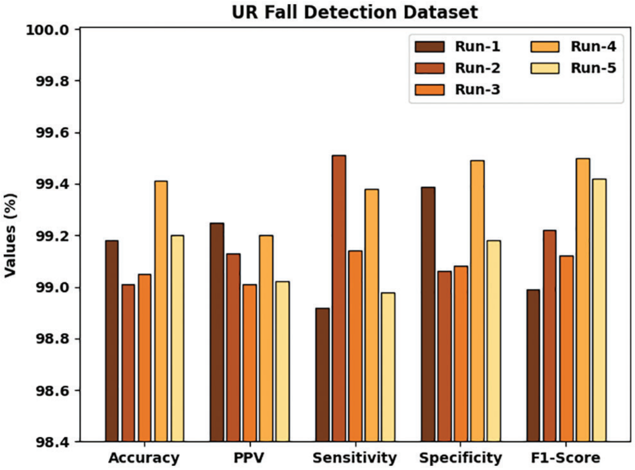

Figure 9: Analytical results of the ADPFD-CSOMN approach on URFD dataset

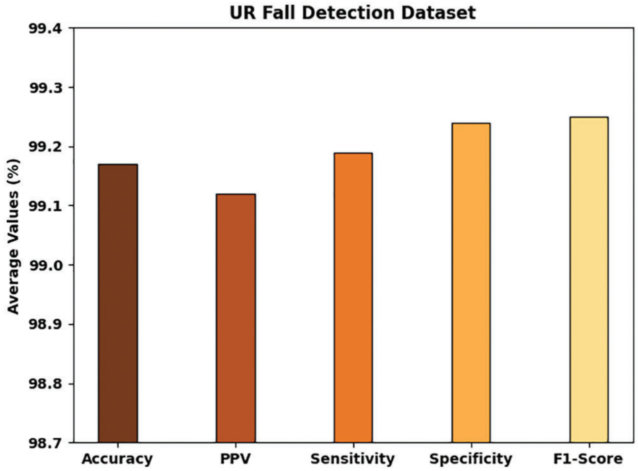

Figure 10: Average analysis results of the ADPFD-CSOMN approach on URFD dataset

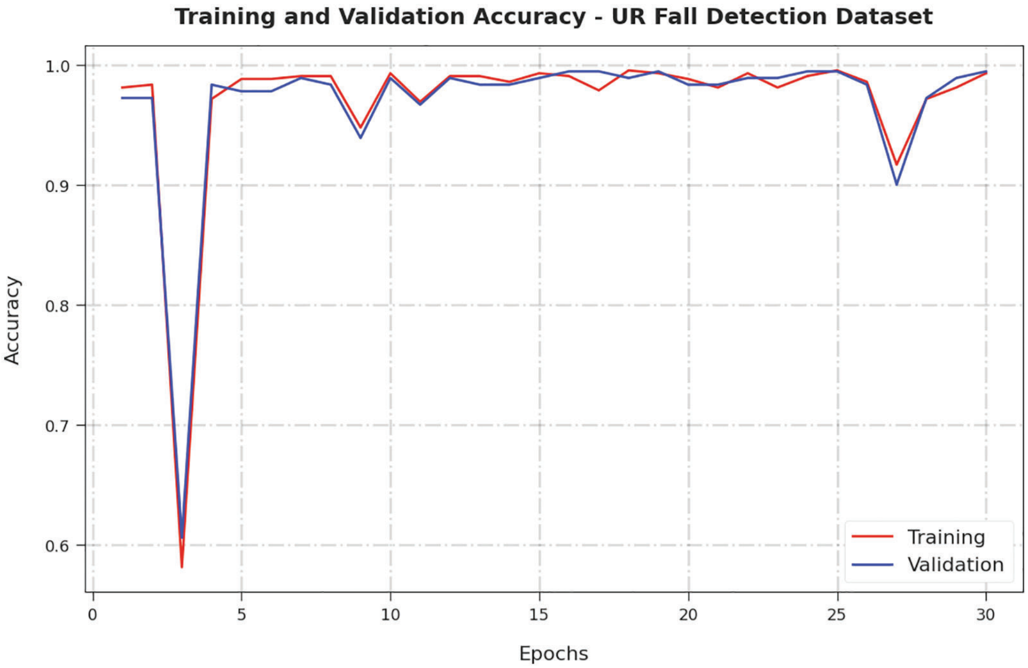

Figure 11: TA and VA analyses results of the ADPFD-CSOMN approach on URFD dataset

Table 5 and Fig. 13 report the comparative analysis results achieved by the ADPFD-CSOMN model and other recent techniques on URFD dataset. The figure infers that the 1D-CNN, 2D-CNN, ResNet-101 and ResNet-50 algorithms reported the least $accu_y$ values such as 92.63%, 94.90%, 95.27% and 95.81% respectively. In the meantime, the depthwise, VGG-16 and VGG-19 models produced reasonable $accu_y$ values such as 98.10%, 97.36% and 98.12% correspondingly. But, the proposed ADPFD-CSOMN method obtained a maximum $accu_y$ of 99.17%.

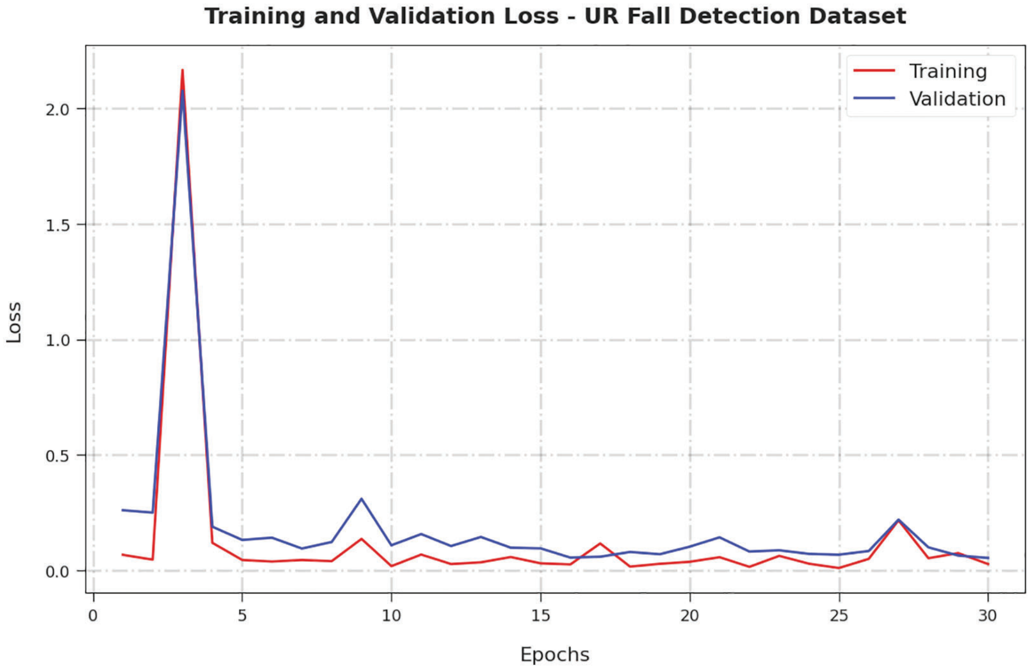

Figure 12: TL and VL analyses results of the ADPFD-CSOMN approach on URFD dataset

Table 5: Comparative analysis results of the ADPFD-CSOMN approach and other recent algorithms on URFD dataset

Methods	Accuracy (%)
ADPFD-CSOMN	99.17
Depthwise algorithm	98.10
ResNet-101 technique	95.27
ResNet-50 technique	95.81
VGG-16 algorithm	97.36
VGG-19 algorithm	98.12
1D-CNN model	92.63
2D-CNN model	94.90

The authors conducted the TRT and TST analyses between the proposed ADPFD-CSOMN model and other recent models on URFD dataset and the results are presented in Table 6 and Fig. 14. Based on the TRT analysis outcomes, it can be inferred that the proposed ADPFD-CSOMN model achieved proficient results with a minimal TRT of 1360 s. However, the depthwise, ResNet-101, ResNet-50, VGG-16, VGG-19, 1D-CNN and the 2D-CNN models produced the maximum TRT values such as 2119, 2673, 3248, 3281, 3206, 1877 and 2628 s correspondingly. In addition, based on the TST analysis outcomes, it can be inferred that the proposed ADPFD-CSOMN model displayed proficient results with a minimal TRT of 696 s. But the depthwise, ResNet-101, ResNet-50, VGG-16, VGG-19, 1D-CNN and 2D-CNN models yielded the maximum TST values such as 1314, 1664, 1668, 1071, 1564, 1494 and 786 s correspondingly.

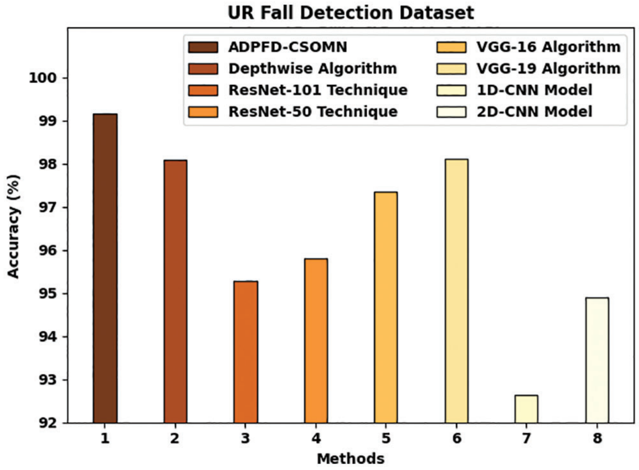

Figure 13: Comparative analysis results of the ADPFD-CSOMN approach on URFD dataset

Table 6: TRT and TST analyses results of the ADPFD-CSOMN approach and other recent algorithms on URFD dataset

	Time (sec)	
Methods	Training	Testing
ADPFD-CSOMN	1360	696
Depthwise algorithm	2119	1314
ResNet-101 technique	2673	1664
ResNet-50 technique	3248	1668
VGG-16 algorithm	3281	1071
VGG-19 algorithm	3206	1564
1D-CNN model	1877	1494
2D-CNN model	2628	786

The outcomes and the discussion made above highlight the proficient performance of the proposed ADPFD-CSOMN model in the fall detection process.

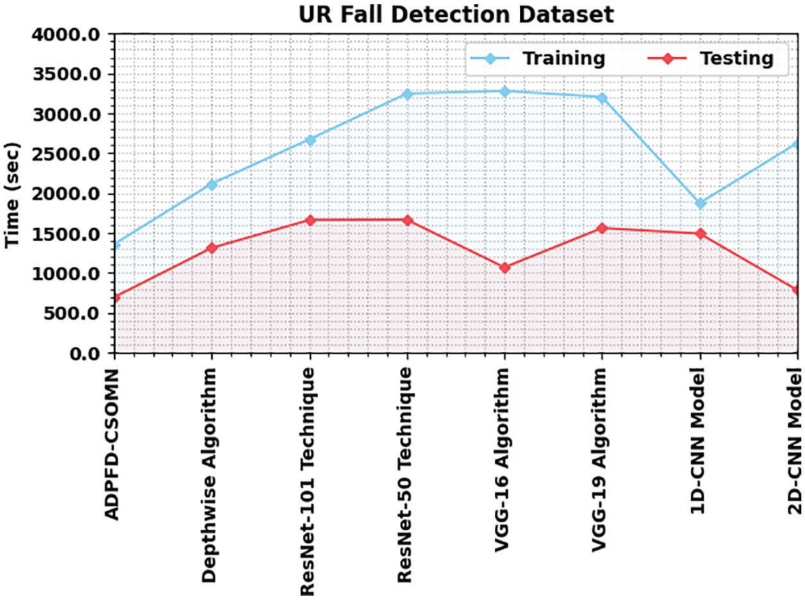

Figure 14: TRT and TST analyses results of the ADPFD-CSOMN approach on URFD dataset

5 Conclusion

In this study, a novel ADPFD-CSOMN method has been devised for identification and the categorization of the fall events. The presented ADPFD-CSOMN technique incorporates the MobileNet model design for the purpose of feature extraction process during the preliminary stage. In addition, the CSO technique is exploited for the optimal modification of the hyperparameters related to the model. In the final stage, the RBF classification model is utilized to recognize and classify an instance as either a fall event or a non-fall event. In order to validate the supremacy of the proposed ADPFD-CSOMN model, a comprehensive experimental analysis was conducted. The comparison study outcomes pointed out the enhanced fall classification outcomes of the ADPFD-CSOMN model over other approaches. Thus, the ADPFD-CSOMN model can be employed as a proficient tool in real-time applications to identify the fall events. In the upcoming years, the performance of the proposed ADPFD-CSOMN model can be optimally improved with the help of hybrid metaheuristic algorithms.

Funding Statement: The author received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] L. M. Villaseñor, H. Ponce, J. Brieva, E. M. Albor, J. N. Martínez *et al.,* “UP-Fall detection dataset: A multimodal approach,” *Sensors*, vol. 19, no. 9, pp. 1988, 2019.

[2] S. Usmani, A. Saboor, M. Haris, M. A. Khan and H. Park, “Latest research trends in fall detection and prevention using machine learning: A systematic review,” *Sensors*, vol. 21, no. 15, pp. 5134, 2021.

[3] S. Rastogi and J. Singh, “A systematic review on machine learning for fall detection system,” *Computational Intelligence*, vol. 37, no. 2, pp. 951–974, 2021.

[4] F. Hussain, F. Hussain, M. E. Haq and M. A. Azam, “Activity-aware fall detection and recognition based on wearable sensors,” *IEEE Sensors Journal*, vol. 19, no. 12, pp. 4528–4536, 2019.

[5] T. Mauldin, M. Canby, V. Metsis, A. Ngu and C. Rivera, “SmartFall: A smartwatch-based fall detection system using deep learning,” *Sensors*, vol. 18, no. 10, pp. 3363, 2018.

[6] A. Bhattacharya and R. Vaughan, “Deep learning radar design for breathing and fall detection,” *IEEE Sensors Journal*, vol. 20, no. 9, pp. 5072–5085, 2020.

[7] I. Chandra, N. Sivakumar, C. B. Gokulnath and P. Parthasarathy, “IoT based fall detection and ambient assisted system for the elderly,” *Cluster Computing*, vol. 22, no. S1, pp. 2517–2525, 2019.

[8] F. Shu and J. Shu, “An eight-camera fall detection system using human fall pattern recognition via machine learning by a low-cost android box,” *Scientific Reports*, vol. 11, no. 1, pp. 2471, 2021.

[9] N. Thakur and C. Han, “A study of fall detection in assisted living: Identifying and improving the optimal machine learning method,” *Journal of Sensor and Actuator Networks*, vol. 10, no. 3, pp. 39, 2021.

[10] F. Lezzar, D. Benmerzoug and I. Kitouni, “Camera-based fall detection system for the elderly with occlusion recognition,” *Applied Medical Informatics*, vol. 42, no. 3, pp. 169–179, 2020.

[11] D. Yacchirema, J. S. de Puga, C. Palau and M. Esteve, “Fall detection system for elderly people using IoT and ensemble machine learning algorithm,” *Personal and Ubiquitous Computing*, vol. 23, no. 5–6, pp. 801–817, 2019.

[12] M. Saleh and R. L. B. Jeannes, “Elderly fall detection using wearable sensors: A low cost highly accurate algorithm,” *IEEE Sensors Journal*, vol. 19, no. 8, pp. 3156–3164, 2019.

[13] M. M. Hassan, A. Gumaei, G. Aloi, G. Fortino and M. Zhou, “A smartphone-enabled fall detection framework for elderly people in connected home healthcare,” *IEEE Network*, vol. 33, no. 6, pp. 58–63, 2019.

[14] N. Zerrouki and A. Houacine, “Combined curvelets and hidden Markov models for human fall detection,” *Multimedia Tools and Applications*, vol. 77, no. 5, pp. 6405–6424, 2018.

[15] J. Maitre, K. Bouchard and S. Gaboury, “Fall detection with uwb radars and cnn-lstm architecture,” *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 4, pp. 1273–1283, 2021.

[16] P. N. Srinivasu, J. G. SivaSai, M. F. Ijaz, A. K. Bhoi, W. Kim *et al.,* “Classification of skin disease using deep learning neural networks with mobilenet v2 and lstm,” *Sensors*, vol. 21, no. 8, pp. 2852, 2021.

[17] J. Yu, C. H. Kim and S. B. Rhee, “Clustering cuckoo search optimization for economic load dispatch problem,” *Neural Computing and Applications*, vol. 32, no. 22, pp. 16951–16969, 2020.

[18] C. She, Z. Wang, F. Sun, P. Liu and L. Zhang, “Battery aging assessment for real-world electric buses based on incremental capacity analysis and radial basis function neural network,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3345–3354, 2020.

[19] T. Vaiyapuri, E. L. Lydia, M. Y. Sikkandar, V. G. Díaz, I. V. Pustokhina *et al.,* “Internet of things and deep learning enabled elderly fall detection model for smart homecare,” *IEEE Access*, vol. 9, pp. 113879–113888, 2021.